WDL 2007 - III Workshop on Digital Libraries, 2007, Gramado, RS.
http://www.archive.org/details/dourado2007_paper_representacao-unificada-odc_wdl2007

Representação Unificada de Objetos Digitais Complexos: confrontando o RAS com o IMS CP

Patrícia Dourado
UNIFACS
Rua dos Colibris, 18
Salvador – BA – Brasil
+55 (71) 3232-4519
patydourado@gmail.com

Pedro Ferreira
UNIFACS
Rua dos Colibris, 18
Salvador – BA – Brasil
+55 (71) 3232-4519
pedrff@gmail.com

André Santanchè
UNIFACS
Av Cardeal da Silva, 747
Salvador – BA – Brasil
+55 (71) 3230-9789
santanche@unifacs.br

RESUMO

A possibilidade de se compartilhar e reusar conteúdo digital na forma de objetos digitais complexos tem motivado pesquisas em diversos domínios, resultando na definição de padrões especializados (multimídia, Educação, bibliotecas digitais e desenvolvimento de *software*) na maioria dos casos, incompatíveis entre si. Esta fragmentação trará problemas a médio prazo, derivados da inevitável necessidade de integração entre domínios. No intuito de se criar um repositório de objetos digitais complexos que integre diversos padrões, iniciou-se um projeto – cujos resultados parciais se apresentam neste artigo – objetivando confrontar os diferentes padrões disponíveis, na busca de soluções de integração entre os mesmos. Partiu-se de um estudo comparativo de dois padrões para a representação de objetos digitais complexos: na área de Educação – Objetos de Aprendizagem – e em desenvolvimento de software – *Reusable Asset Specification* (RAS). Para realizar o comparativo fez-se um estudo de caso prático utilizando ferramentas que adotam tais padrões.

Descritores de Categoria e Assunto

H.3.7 [**Information Storage and Retrieval**]: Digital Libraries – *dissemination, standards.*

Termos Gerais

Padronização.

Palavras-Chave

Objetos Digitais Complexos; *Reusable Asset Specification*; Objetos de Aprendizagem.

1. INTRODUÇÃO

A evolução da Web e o aumento do poder computacional incrementaram a capacidade de produção e compartilhamento do conteúdo digital, que vai se tornando cada vez mais complexo. Para representar a complexidade deste conteúdo nasceu, no contexto das bibliotecas digitais, a noção "objeto digital complexo" [1] ou simplesmente "objeto complexo".

Nos últimos anos tem crescido o interesse pelo tema de objetos complexos em muitos domínios de aplicação, dando origem a diversos padrões para representação, empacotamento e distribuição dos mesmos: o IMS CP (*IMS Content Package*) [11] para conteúdo educacional, o RAS (*Reusable Asset Specification*) [9] para artefatos gerados no ciclo de desenvolvimento de software, MPEG-21 (*Moving Picture Experts Group –21*) [2] para artefatos multimídia e METS (*Metadata Encoding and Transmission Standard*) [12] voltados ao domínio de bibliotecas digitais.

Uma vez que cada padrão é especializado em um domínio específico, as diferenças entre eles e os seus formatos não permite a interoperabilidade entre os mesmos. No entanto, quando se trata de criar bibliotecas genéricas de objetos complexos, torna-se necessário oferecer suporte a uma grande variedade de domínios. Neste caso, a escolha de um padrão específico pode significar a exclusão de objetos compartilhados em outros padrões.

Em um estudo anterior realizado em [10], observou-se que os principais aspectos que caracterizam um objeto complexo são representados de forma equivalente nos diferentes padrões, o que torna viável a construção de um modelo para integrá-los.

No ano de 2006 iniciou-se o projeto *MediaBank* cujo objetivo é a construção de um repositório que será capaz de armazenar diversos tipos de mídia. Dada a diversidade e heterogeneidade de mídias projetadas para armazenamento – tanto na versão atual quanto em futuras expansões –, optou-se pela representação do conteúdo na forma de objetos digitais complexos. Isto motivou o desenvolvimento da pesquisa apresentada neste artigo, que constitui um subconjunto do projeto *MediaBank*. Seu propósito é a criação de um modelo que permita integrar as diversas perspectivas de objetos complexos, através de um novo padrão de objetos complexos que generalize os demais, e que será armazenado em um repositório unificado, conforme mostrado na Figura 1.

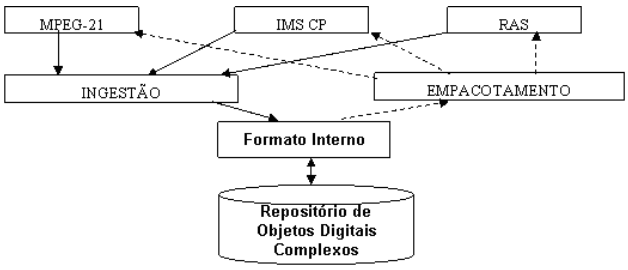

Figura 1 - Modelo de integração entre padrões de objetos complexos em repositório unificado.

A metodologia deste trabalho se baseia no estudo aprofundado de padrões de objetos complexos já existentes, tais como: *IMS CP, RAS, METS, MPEG –21*, a fim de identificar similaridades e diferenças, que serão a base do nosso modelo integrado. Em seguida, será implementado um sistema de gerenciamento de um repositório de objetos complexos, bem como um mecanismo de conversão dos demais padrões já existentes para o padrão proposto, e vice-versa.

Apresenta-se aqui a primeira etapa desta metodologia, que é o estudo comparativo entre os padrões. Inicialmente foram escolhidos dois dos padrões – IMS CP e RAS – para estudo, e pretende-se estender os resultados obtidos aos demais padrões. O IMS CP é um modelo para empacotamento de objetos complexos educacionais, definido pelo *IMS Global Learning Consortium* (http://www.imsproject.org), e o RAS, que foi proposto pela OMG (*Object Management Group*), tem o objetivo de definir um padrão de empacotamento e descrição de artefatos digitais usados no processo de desenvolvimento de *software*. A comparação entre os padrões apresentada neste artigo foi feita em duas dimensões: a comparação teórica entre as especificações, e a comparação prática utilizando um estudo de caso com ferramentas que implementam ambos os padrões. No estudo de caso, o mesmo objeto digital complexo foi formatado nos padrões IMS CP e RAS para comparação.

O presente artigo está organizado em quatro seções. Na Seção 2 propõe-se o estudo de caso e como ele foi aplicado aos padrões. Na Seção 3 apresenta-se uma revisão de trabalhos relacionados na área de objetos digitais complexos. Na Seção 4 faz-se uma análise comparativa dos padrões. Na Seção 5 expõem-se as considerações finais do trabalho.

2. ESTUDO DE CASO

Este estudo de caso adotou o exemplo de um objeto digital complexo de conteúdo educacional.

2.1 Conteúdo Educacional em Biologia

Considere que um professor de biologia necessita de um material educativo sobre o corpo humano, o qual apresente alguns de seus órgãos e suas funcionalidades. Para isso, criou-se um objeto complexo, concebido para ser executado em um navegador Web. A Figura 2 apresenta a tela inicial deste objeto complexo no momento em que ele é apresentado em um navegador Web. O aluno pode clicar em partes do corpo humano e, através de links mapeados na imagem, será conduzido a outras páginas que contêm detalhes sobre o respectivo órgão. Para fins de simplificação, apenas uma parte do objeto, detalhando o coração e pulmão, foi incluída no exemplo apresentado.

Figura 2 - Página inicial do exemplo.

Este objeto digital complexo é composto de páginas HTML, imagens, folhas de estilos, um arquivo PDF. Esses arquivos de naturezas distintas e as dependências existentes entre eles ilustram a complexidade típica de muitos objetos digitais.

A Figura 3 apresenta um diagrama de dependências de artefatos dentro do objeto complexo. Pode-se observar uma página HTML no topo, que: utiliza uma imagem de um coração, define um link que referencia outra página HTML (que explica mais detalhadamente as funções de artérias e veias), e define um link para um arquivo PDF. Ë necessário identificarem-se as dependências existentes entre os diversos artefatos para garantir a integridade do conteúdo e permitir o seu reuso. A página "coracao.html", por exemplo, depende da imagem de um coração para que ela seja apresentada da maneira correta. O controle refinado das dependências torna-se mais importante quando se pretende reusar um subconjunto do objeto complexo, por ser necessária a garantia de que o subconjunto extraído contenha todos os artefatos necessários para o seu correto funcionamento.

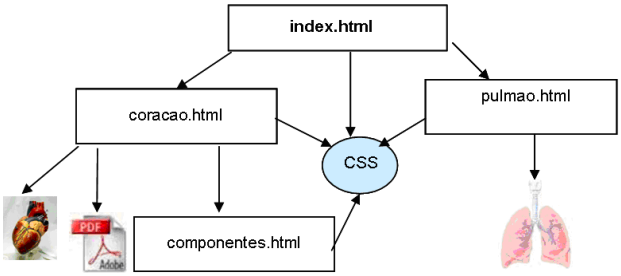

Figura 3 - Diagrama dos artefatos que compõem o objeto complexo do estudo de caso e suas dependências.

Neste exemplo, foram propositalmente escolhidos objetos de diferentes naturezas com o propósito de enriquecer a análise apresentada a seguir.

2.2 Ferramentas de implementação

Na tarefa de implementação do estudo de caso adotaram-se duas ferramentas, uma para cada um dos padrões. Para o IMS CP foi utilizado o *Reload Editor* e para o RAS foi usado o Rational XDE.

O *Reload Editor* é uma ferramenta utilizada para organização, anotação e empacotamento de objetos de aprendizagem, que faz parte do projeto *Reusable eLearning Object Authoring & Delivery* (RELOAD – http://www.reload.ac.uk/). Embora o *Reload Editor* apresente diversas maneiras para a criação de estruturas de objetos de aprendizagem, neste exemplo foi utilizada a estrutura de criação para *IMS Content Package*.

O *Rational XDE* da IBM (http://www.ibm.com/software/awdtools/developer/rosexde/) é um ambiente integrado de projeto e desenvolvimento de software, que funciona dentro do ambiente Eclipse (http://www.eclipse.org/). Entre as suas funcionalidades, ele é capaz de empacotar um projeto em formato RAS. Assim como o *Reload Editor*, o *Rational XDE* possui diversas opções para a criação da estrutura do arquivo RAS, mas para este exemplo foi utilizada a estrutura de manifesto *Default* (analisada adiante) e o tipo de exportação do arquivo através da opção "*Export Whole Projects*".

3. TRABALHOS RELACIONADOS

Nesta seção serão apresentados trabalhos relacionados a objetos digitais complexos, com enfoque em dois padrões: IMS CP – (*IMS Content Package*) [11] – e RAS – (*Reusable Asset Specification*) [9]. Aqui se detalhará como a mesma implementação do estudo de caso da seção anterior será representada tanto em IMS CP quanto em RAS, proporcionando condições equivalentes de testes para ambos. Tais implementações serão a base para a comparação apresentada na seção subseqüente.

3.1 IMS CP

Objetos complexos são usualmente armazenados dentro de estruturas de pacote. Um pacote é uma estrutura responsável pela agregação de múltiplos artefatos digitais em um arquivo único, preservando sua organização.

A estrutura de um pacote de IMS CP é definida conforme a Figura 4. Na figura, a estrutura externa, indicada como *Package*, consiste em um arquivo compactado em formato ZIP. Dentro do ZIP, está um arquivo especial denominado manifesto (*manifest*) com quatro subdivisões. Este arquivo será comentado adiante. Além do manifesto, ficam dentro do ZIP todos os demais artefatos que compõem o objeto complexo.

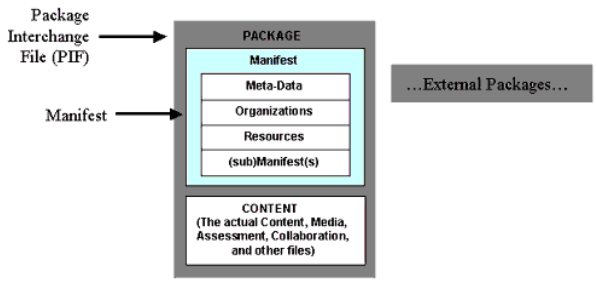

Figura 4. Estrutura de um pacote IMS CP [11].

O formato ZIP não apenas reúne todos os artefatos em um único arquivo, como é capaz de reproduzir a estrutura hierárquica dos diretórios que compõem o objeto. A Figura 5 apresenta o pacote IMS CP aberto por um programa de manipulação de arquivos ZIP. Como é possível observar, o formato IMS CP utiliza o formato ZIP sem modificações. Todas as informações adicionais relativas aos artefatos se concentram no arquivo de manifesto, que aparece na raiz do arquivo compacto (imsmanifest.xml).

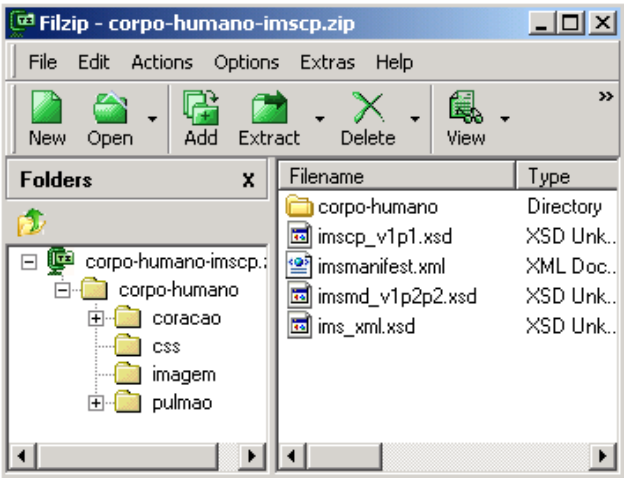

Figura 5 - Objeto de Aprendizagem em formato ZIP.

Na Figura 6, foi capturada uma tela do *Reload Editor* durante o processo de edição de um pacote IMS CP. À esquerda aparece a organização dos artefatos dentro do arquivo ZIP, à direita é detalhado o arquivo de manifesto. O destaque para o arquivo de manifesto ressalta a sua importância como estrutura complementar para a organização dos artefatos.

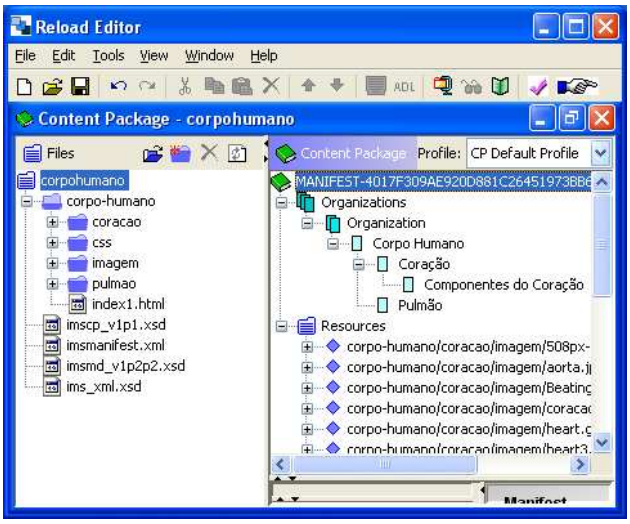

Figura 6 - *Reload Editor* para criação IMS CP.

No padrão IMS CP o manifesto é dividido nas seguintes seções (Figura 4):

Meta-Data – metadados educacionais conforme o padrão *Learning Object Metadata* (LOM) [4] que descrevem o objeto como um todo.

Organização – define uma estrutura organizacional hierárquica que funciona como um índice de tópicos e subtópicos associados ao conteúdo educacional.

Recursos – contém referências para os artefatos que estão armazenados no ZIP e mapeia as dependências entre eles.

Submanifestos – esta seção é opcional, e contém manifestos subordinados, quando há pacotes dentro de pacotes.

3.2 RAS

A Figura 7 detalha a estrutura do arquivo de extensão RAS, definido pela OMG [9]. Tal estrutura é equivalente à apresentada na Figura 4. Como pode ser observado na figura, do mesmo modo que no IMS CP, um objeto RAS é encapsulado dentro de um arquivo em formato ZIP, e define um arquivo adicional de manifesto.

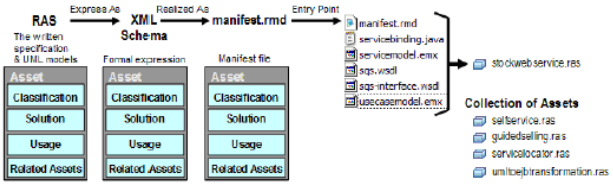

Figura 7 - Formato de arquivo RAS [9].

A fim de comparar a estrutura das duas abordagens (IMS CP e RAS), como ilustrado na Figura 8, o arquivo RAS gerado pelo *Rational XDE* também foi aberto pela mesma ferramenta de manipulação de arquivos ZIP da Figura 5. Note-se que, neste caso, o arquivo de manifesto (rasset.xml) também é armazenado na raiz do arquivo compacto.

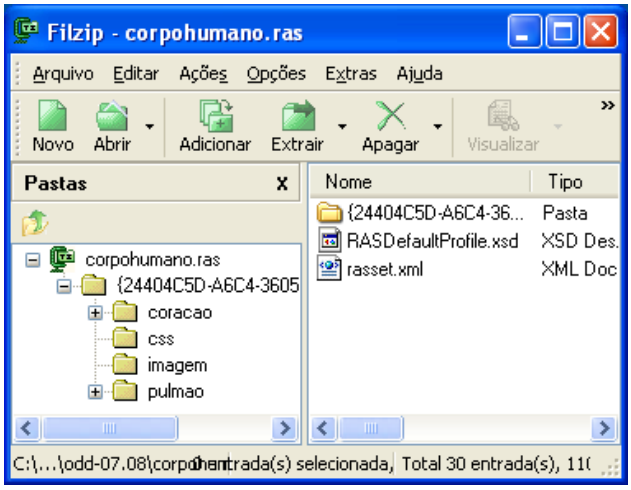

Figura 8 - Arquivo .ras em formato ZIP.

Como está ilustrado na Figura 7, o arquivo de manifesto RAS também possui uma estrutura interna com subdivisões, que deriva de um modelo UML, conforme apresentado na Figura 9. Na figura, a classe *Asset* representa o núcleo comum de representação de todos os manifestos. Cada classe derivada de *Asset*, caracterizada como *Profile*, pode representar um manifesto específico para um tipo de objeto complexo. O *Default Profile* é um manifesto genérico, aplicável a objetos complexos RAS. O *Default Component Profile* e o *Default Web Service Profile* são especializações do *Default Profile* para componentes e serviços Web respectivamente [9].

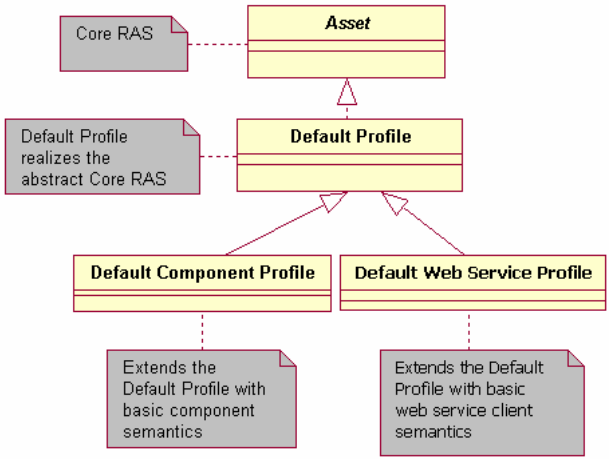

Figura 9 - Modelo UML do manifesto RAS [9].

O padrão RAS estabelece uma estratégia de mapeamento de cada classe *Profile* definida no modelo para o respectivo esquema XML. Esta estratégia estimula a definição de extensões de forma organizada, documentada e de fácil implementação.

A Figura 10 detalha a classe Asset apresentada na Figura 9, e apresenta o modelo principal das seções do manifesto RAS, cada uma representada por uma classe.

Classification – lista os descritores usados para classificar o objeto complexo, bem como faz a descrição do contexto para o qual ele é importante.

Solution – descreve os artefatos que compõem o objeto complexo.

Usage – contém as regras para instalação, customização e uso do objeto complexo.

Related Asset – descreve o relacionamento entre os objetos complexos.

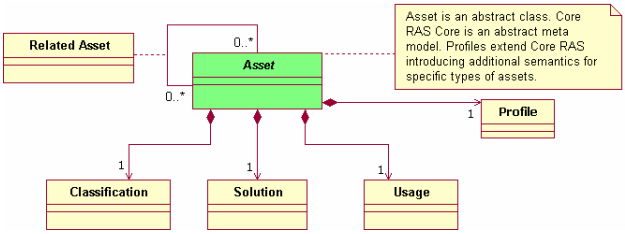

Figura 10 - Modelo UML detalhando a classe *Asset* [9].

4. ANÁLISE COMPARATIVA

Este trabalho tem como ponto de partida um estudo anterior, realizado por Santanchè [10], no qual foram comparados os quatro padrões de representação objetos digitais complexos, referenciados na Seção 1 – IMS CP, RAS, MPEG-21 [2] e METS [12] –, além de padrões usados para a representação de componentes de *software* – CCM (CORBA Component Model), Microsoft COM+ e Sun EJB (Sun Enterprise Java Beans). A comparação entre padrões de objetos complexos e componentes de software se dá pelo fato que Santanchè propõe uma

combinação de ambas as abordagens em um modelo unificado, o Componente de Conteúdo Digital (Digital Content Component – DCC).

Em Santanchè [10] foi feita uma comparação inicial e foram apontadas algumas diretrizes para a combinação dos modelos utilizados por objetos complexos. Este trabalho aprofundou as diretrizes iniciais, realizou estudos de caso práticos sobre o tema e está desenvolvendo as bases para a construção de mapeamentos entre os padrões disponíveis.

Na seção anterior foram ressaltadas algumas similaridades estruturais entre o IMS CP e o RAS. Como foi observado por [10], e está ilustrado na Figura 11, os formatos de empacotamento de diversos padrões objetos digitais complexos têm equivalências estruturais.

Cada círculo preenchido em cinza, no centro da Figura 11, representa um formato de empacotamento adotado por um padrão de objeto complexo ou de componente de software. O padrão OAIS XFDU é um formato que está sendo proposto para o empacotamento de objetos que têm metadados conforme o padrão METS. As linhas pontilhadas indicam que o padrão ainda estava em fase de definição no momento da realização da pesquisa.

Como foi analisado na Seção 3, os padrões de representação de objetos complexos combinam um formato de pacote, que na figura está indicado como "*Package Container*", e um formato de manifesto, que na figura está indicado como "*Manifest*". Os círculos em branco indicam formatos de empacotamento ou manifesto e as setas representam a derivação entre os formatos.

Como mostra a figura, a maioria dos padrões optou pelo pacote em formato ZIP, ou ainda não definiu o formato. Todos guardam dentro do pacote um arquivo de manifesto em formato XML, com o propósito de representar dados complementares de organização, dados específicos de domínio e metadados. As diferenças entre os diversos padrões se concentram principalmente na representação deste arquivo de manifesto. Por este motivo se concentrará na comparação dos manifestos, para estabelecer as diferenças e similaridades entre IMS CP e RAS.

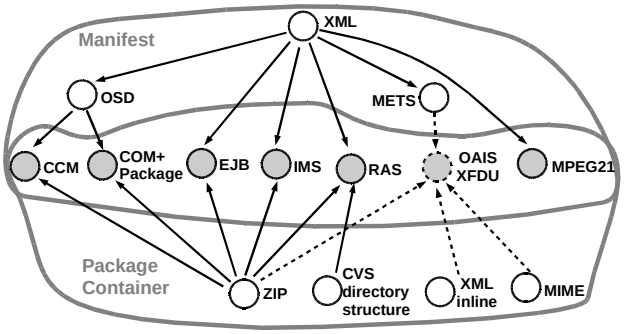

Figura 11 - Comparação das estruturas de empacotamento [10].

As estruturas de manifesto são compostas por diversos elementos, nos quais alguns são independentes do domínio do objeto complexo e outros são bastante especializados. Aqueles independentes de domínio podem ser facilmente generalizados e mapeados entre diferentes padrões. Neste artigo se apresenta o confronto daqueles elementos independentes de domínio. Devido à limitação de espaço, escolheram-se alguns destes elementos considerados os mais relevantes para a análise realizada.

4.1 Organização

Como foi apresentado na seção anterior, o IMS CP faz uso da seção *Organizations* que organiza hierarquicamente o conteúdo em tópicos e subtópicos. Esta organização foi planejada para ser usada por ferramentas educacionais que irão apresentar o conteúdo. Por exemplo, o professor pode disponibilizar o objeto complexo em formato IMS CP para seus alunos em um Ambiente Virtual de Aprendizagem (AVA), tal como o Moodle (http://moodle.org/), conforme o exemplo da Figura 12. O professor importa diretamente o pacote compacto IMS CP no ambiente e o conteúdo será apresentado, mantendo a organização definida na seção *Organization*. Note-se que o índice de organização apresentado à esquerda na Figura 12 – com *links* para páginas específicas do conteúdo – é o mesmo apresentado na Figura 6.

Figura 12 - Pacote IMS CP apresentado no Moodle.

No site do MIT *OpenCourseWare* (http://ocw.mit.edu/index.html) é possível encontrar recursos gratuitos de diversas áreas para estudantes e professores. Os recursos são organizados e empacotados conforme o padrão IMS CP. Adicionalmente o *OpenCourseWare* definiu uma estrutura padronizada para a hierarquia de organização, o que facilita a navegação em diferentes objetos.

O RAS não possui uma estrutura equivalente, uma vez que ele não foi concebido especificamente para conteúdo educativo. Entretanto, o padrão poderia beneficiar-se desta estrutura para artefatos de documentação de software, por exemplo.

4.2 Referência aos Recursos/Artefatos

A referência a artefatos ou recursos digitais é feita de maneira equivalente no IMS CP e RAS com diferenças nas nomenclaturas.

No padrão IMS CP o artefatos são descritos na seção *Resources*. A cada artefato é atribuído um identificador e uma relação de outros artefatos, dos quais este artefato é dependente. No padrão RAS, os artefatos são descritos na seção *Solution*. Para cada artefato é criado um identificador, que será utilizado para relacionar a dependência de um artefato com outros artefatos.

4.3 Identificação

O esquema de identificação recomendado pelo IMS se baseia no *Uniform Resource Name* (URN) [7], que consiste em um esquema de identificação usado na Web para a produção de identificadores persistentes e globalmente únicos [8].À diferença das conhecidas

URLs, os URNs ou requerem a existência de um servidor cujo papel é resolver nomes, ou seja, converter os URNs em localizações físicas na rede. Isto garante que a localização física do recurso identificado seja encontrada indiretamente, permitindo que a URN se mantenha mesmo quando existem reorganizações físicas na rede.

O RAS faz uso do *Universally Unique Identifier* (UUID) para identificação do objeto complexo [5]. O UUID foi padronizado pela *Open Software Foundation* (OSF), e uma das vantagens da sua utilização é que não requer um órgão centralizado para administrar a geração desses identificadores únicos. Este identificador é um número representado em hexadecimal, com tamanho fixo de 128 bits, como por exemplo: uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6. Para gerar este número são usados algoritmos, que podem basear-se em identificadores de hardware, geração de números aleatórios e códigos hashing calculados a partir do objeto.

Uma das vantagens na utilização do UUID em relação ao uso do URN é o fato de este identificador não necessitar de um órgão central para administrar, o que gera uma independência em relação à rede e a servidores. Por outro lado, o URN é um mecanismo fundamental de identificação na Web Semântica; seu uso pode facilitar a interligação de metadados com descrições RDF e ontologias OWL.

4.4 Perfil e Extensão

A concepção do modelo RAS tem uma característica distintiva em relação ao IMS CP. Ao invés de se concentrar apenas na construção de um esquema para o manifesto, o RAS partiu de um modelo, representado em UML, que derivou para o esquema. Como foi apresentado, o RAS possui uma família de esquemas (*profiles*) que se relacionam através de uma cadeia de derivações (herança em UML). Esta cadeia baseia-se na utilização do id do *Profile*. Um *Profile*, além de definir a estrutura e semântica do documento manifesto do objeto complexo (tal como uma classe UML), pode ter diferentes versões e ser usado como referência para outros *Profiles*. A identificação do *Profile* irá registrar toda a cadeia de derivação do esquema RAS utilizado. Um esquema RAS pode ser derivado a partir de Core *Profile* ou a partir de qualquer derivação do *Profile*. Na aplicação da derivação somente poderão ser adicionados elementos e atributos ao esquema XML do manifesto, e/ou associar nova semântica aos elementos existentes. A derivação do *Profile* jamais poderá remover elementos e atributos do esquema XML original. A Figura 13 apresenta um recorte do arquivo de manifesto RAS do estudo de caso, no qual aparece o atributo id-history. Este atributo é uma chave composta que identifica o id do *Profile* original e a seqüência de ids de derivação separados por duplos dois-pontos (::). O último id da seqüência é o do *Profile* usado no manifesto.

Os atributos *version-major* e *version-minor* ajudam a identificar a versão do *Profile*. Eles são usados para distinguir *Profiles* com nomes iguais.

```
<profile id-history="F1C842AD-CE85-4261-ACA7-178C457018A1
::31E5BFBF-B16E-4253-8037-98D70D07F35F"
name="Default" version-major="2" version-minor="01" />
```

Figura 13. Derivação Core Profile.

Quanto aos objetos complexos, O RAS provê um atributo opcional *version*, para controle de versão do artefato. O padrão IMS CP utiliza um controle de versão opcional, que pode ser descrito no atributo *version*.

4.5 Tipos de Artefatos

O IMS CP possui um critério limitado para reconhecimento de tipos de artefatos. A última especificação disponível indica três tipos de artefato apenas. Conseqüentemente, o *Reload Editor* identificou todos os artefatos como "*webconten*t", independente do modelo de objeto de aprendizagem ser criado para o domínio WEB ou não. Inserimos no estudo de caso na Seção 2 um arquivo executável não compatível com navegadores Web, simulando os batimentos do coração. A Figura 14 apresenta um recorte do manifesto IMS CP desta variante. Note-se que, apesar de não ser criado para o domínio Web, o artefato é identificado e caracterizado pelo padrão do IMS CP, como um tipo *type="webcontent*".

```
- <resource identifier="RES-B23574EDF53173AE46F9CAD3CD97BC84"
type="webcontent" href="anima/message/java/AnimaEventSource.class">
  <file href="anima/message/java/AnimaEventSource.class" />
  </resource>
```

Figura 14 - Definição de tipo arquivo executável no IMS CP.

No RAS, o mesmo exemplo é destacado na Figura 15 e pode-se observar que o atributo "type" do manifesto gerado está associado ao tipo correto: tipo Java.

O padrão RAS define um conjunto mais rico de tipos, o que dá mais opções para a ferramenta que produz o manifesto.

```
<artifact id="{85431381-E72C-449F-C1B4-F2DAA5730366}"
name="MessageSingle"reference="{97B2A319-4B54-2A7D-ED21-D1A742949479}
\anima\message\MessageSingle.java" type="Java" />
```

Figura 15 - Declaração de tipo em arquivo executável no RAS.

Os padrões definem tipos próprios para identificar os artefatos, o que pode resultar em problemas de interoperabilidade.

5. CONSIDERAÇÕES FINAIS

Os dois padrões detalhados nas seções anteriores – IMS CP e RAS – deixam perceber que, embora cada padrão possua características de empacotamento de recursos específicos, e como conseqüência, estruturas distintas para a apresentação dos mesmos, existem muitas características compatíveis que permitem um mapeamento direto entre os padrões. Adicionalmente, cada um dos padrões apresenta vantagens que podem se somar em uma abordagem unificada de representação.

Este estudo forneceu subsídios para a caracterização dos requisitos de um modelo unificado de representação de objetos digitais complexos. Além disso, está clara a necessidade de se estenderem ambos os padrões para que seja possível o mapeamento entre eles.

A estratégia a ser usada na elaboração dos mapeamentos se baseia no uso de folhas XSLT para realizar transformações. Para a identificação dos recursos será usada uma abordagem que unifica UUIDs com URNs [6].

Além disto, estão sendo produzidas extensões no modelo RAS objetivando oferecer suporte a conteúdo educacional, que contenha pelo menos todos os metadados disponíveis no IMS CP. A extensão se baseia na criação de um *Profile* para objetos de aprendizagem, como está ilustrado na Figura 16.

UUID A::UUID B

UUID A::UUID C::UUID D

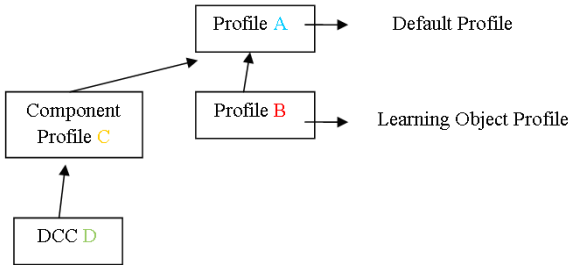

Figura 16 - Extensão RAS para Objetos de Aprendizagem.

O *Learning Object Profile* irá estender o *Default Profile*, mantendo as características iniciais do mesmo e adicionando elementos e/ou semântica dos Objetos de Aprendizagem.

A extensão no sentido inverso também é possível, mas tem-se mostrado uma tarefa mais complexa, que envolve a extensão do esquema em pontos pré-definidos.

O estudo desenvolvido neste artigo se concentrou no IMS CP e RAS, mas utilizou uma abordagem suficientemente genérica para que ele possa ser estendido a outros padrões citados, tais como o MPEG21 e METS.

Agradecimentos

Este trabalho foi parcialmente financiado pelos projetos LabMultflex (FAPESB-Infra 2006), MediaBank (HP Digital Publishing) e WebMaps II (CNPq).

REFERÊNCIAS

[1] Bekaert, J. , Kooning, E. D., Walle, R. V. de. *Packaging models for the storage and distribution of complex digital objects in archival information systems: a review of MPEG-21 DID principles*. ***Multimedia Systems,*** p. 286–301, abr. 2005.

[2] Burnett, I . et al. MPEG-21: *Goals and Achievements*. IEEE **MultiMedia,** p. 60-70, out–dez. 2003.

[3] CCSDS. XML *Formatted Data Unit (XFDU) Structure and ConstructionRules*, set. 2004. <http://sindbad.gsfc.nasa.gov/xfdu/pdfdocs/iprwbv2a.pdf>, Acesso em: set. 2007.

[4] IEEE L.T.S.C. *Draft Standard for Learning Object Metadata – IEEE 1484.12.1*, 2002. <http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf >. Acesso em: set. 2007.

[5] ISO/IEC 9834-8:2004 Information Technology. *Procedures for the operation of OSI Registration Authorities: Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier components, ITU-T Rec. X.667*, 2004.

[6] Leach, P., Mealling, M., Salz, R. *A Universally Unique IDentifier (UUID) URN Namespace – IETF RFC 4122*, jul. 2006.

[7] McKell, M. IMS Persistent, Location-Independent, Resource Identifier Implementation Handbook, abr. 2001. Disponível em: <http://www.imsproject.org/implementationhandbook/imsrid_handv1p0.html>. Acesso em: ago. 2007.

[8] Moats R. URN Syntax. Network Working Group, mai. 1997.

[9] OMG. Reusable Asset Specification. Final specification – version 2.2, April 2005. Disponível em: <http://www.omg.org/docs/formal/05-11-02.pdf >. Acesso em: jan. 2007.

[10] Santanchè, A. and Medeiros, C. B. A Component Model and Infrastructure for a Fluid Web. IEEE Transactions on Knowledge and Data Engineering, 19(2):324–341, 2007.

[11] Smythe, C., Jackl, A. (eds.). IMS Content Packaging Information Model. Specification, IMS Global Learning Consortium, Inc., October 2004. Disponível em: <http://www.imsglobal.org/content/packaging/cpv1p1p4/imscp_infov1p1p4.html>. Acesso em: mar. 2007.

[12] The Library of Congress. METS: An Overview & Tutorial, set. 2004. Disponível em: <www.loc.gov/standards/mets/METSOverview.v2.html>. Acesso em: mar. 2007.